Rhythmic-Motion Synthesis Based on Motion-Beat Analysis

Tae-hoon Kim Sang Il Park Sung Yong Shin

Korea Advanced Institute of Science and Technology

Abstract

Real-time animation of human-like characters is an active research area in computer graphics. The conventional approaches have, however, hardly dealt with the rhythmic patterns of motions, which are essential in handling rhythmic motions such as dancing and locomotive motions. In this paper, we present a novel scheme for synthesizing a new motion from unlabelled example motions while preserving their rhythmic pattern. Our scheme first captures the motion beats from the example motions to extract the basic movements and their transitions. Based on those data, our scheme then constructs a *movement transition graph* that represents the example motions. Given an input sound signal, our scheme finally synthesizes a novel motion in an on-line manner while traversing the motion transition graph, which is synchronized with the input sound signal and also satisfies kinematic constraints given explicitly and implicitly. Through experiments, we have demonstrated that our scheme can effectively produce a variety of rhythmic motions.

CR Categories: K.6.1 [Management of Computing and Information Systems]: Project and People Management—Life Cycle; K.7.m [The Computing Profession]: Miscellaneous—Ethics

Keywords: Beat Analysis, Motion Synchronization, Motion Synthesis, Motion Blending, Motion Transition, Motion Signal processing,

1 Introduction

Synthesizing realistic human motions is an important issue in computer animation. The advent of motion capture devices has greatly facilitated the acquisition of realistic human motion data and attracted many researchers to develop motion editing techniques to reuse captured motion clips guided by user-specified constraints [Bruderlin and Williams 1995; Gleicher 1998; Lee and Shin 1999; Popović and Witkin 1999; Rose et al. 1996; Shin et al. 2001; Unuma et al. 1995; Witkin and Popović 1995]. Brand and Hertzmann proposed a statistical model called “style machine” learned from captured motion data, to generate new motions of various styles [Brand and Hertzmann 2000]. Recently, example-based approaches have been explored to synthesize novel motions either by blending labelled motion clips of an identical structure [Park et al. 2002; Rose et al. 1998; Wiley and Hahn 1997] *or* by rearranging unlabelled motion data [Arikan and Forsyth 2002; Galata et al. 2001; Kovar et al. 2002; Lee et al. 2002; Li et al. 2002; Tanco and Hilton 2000].

While mainly focusing on generation of convincing posture sequences, the conventional techniques barely deal with structured, regular temporal patterns called *rhythms* contained in example motions. The rhythms are essential in producing motions such as dancing and marching, which can be regarded as simultaneous responses to external rhythmic signals such as background music and drum beats. To create a new rhythmic motion from example motions, we need to consider the rhythms involved in the example motions as well as the constituent movements themselves.

In this paper, we present a novel approach to synthesizing a new rhythmic motion from unlabelled example motions while maintaining their rhythmic features. To capture the rhythms of example motions, we introduce the notion of a *motion beat* and that of a *rhythmic pattern* based on the results in choreography: The motion beat is a regular, rhythmic unit of time for a motion, and the rhythmic pattern is a sequence of motion beats corresponding to a motion unit [Blom and Chaplin 1982; Hawkins 1988].

Given example motions consisting of variations (examples) of motion units called *basic movements*, we extract their motion beats by detecting the moments of evident directional changes. It is well-known that distinctive directional changes periodically occur in real motions [Jones and Boltz 1989]. However, not all moments of such directional changes are necessarily motion beats. Thus, to identify the motion beats, we estimate the dominant periodic pattern over the temporal distribution of such moments. Every example motion is represented as a sequence of basic movements, each of which spans consecutive motion beats and corresponds to the rhythmic pattern underlying the motion. We identify the basic movements embedded in the example motions by punctuating the corresponding streams of motion beats with *a priori* knowledge on their rhythmic pattern, and then classify them into groups according to their similarities.

Every group of basic movements can be regarded as variations of a prototype movement. We model the prototype movements and their transitions as a *movement transition graph*, of which each node and edge represent a prototype movement and the transition (possibly self-transition) from a prototype movement to a prototype movement, respectively. A node contains a group of basic movements, that is, variants of a prototype movement. By analyzing the connections between basic movements in the example motions, we find the likelihoods of transitions that reflect the natural, organized progression of those motions. By blending the basic movements at a node, we can synthesize a novel basic movement of the same structure. The temporal correspondence among those movements is determined by their motion beats, since all basic movements have the same rhythmic pattern.

Our motion beat analysis scheme facilitates the generation of a novel rhythmic motion synchronized with an input sound signal such as background music, to create convincing animations on the fly. After constructing the movement transition graph, we traverse this graph from node to node guided by the transition probabilities until the sound signal ends, while synthesizing a basic movement at each node. In addition to the temporal constraints given by the sound signal, we also make the resulting motion satisfy the kinematic constraints either given explicitly *or* derived from the environment changing dynamically. Our approach can also be used for controlling a group of characters automatically in virtual environments, for example, characters dancing in a ballroom *and* soldiers marching in formation.

The remainder of this paper is organized as follows: We first review related work in Section 2. We present a novel scheme for extracting the motion beats from rhythmic motions in Section 3. Then, we describe how to represent a set of example motions as a movement transition graph in Section 4. Motion synthesis from the movement transition graph is presented in Section 5. We show experimental results in Section 6. Finally, Sections 7 and 8 provide discussion and conclusion, respectively.

2 Related Work

In computer animation, motions of characters have frequently been combined with music in order to convey various emotions [Laybourne 1998; Pausch et al. 1996]. The synchronization of an animated image sequence with sounds helps deliver an artistic theme to the audience [Laybourne 1998]. However, to our knowledge, there have been few attempts to automatically produce novel motions synchronized with music by reusing existing motion data. Animators have spent their time and efforts in creating new motions, the rhythm of which matches that of a given piece of music [Cassell et al. 2001; Laybourne 1998].

Extracting the temporal structures from rhythmic signals such as drum sounds, music, and motion is a difficult problem. In psychology, the human ability to perceive and produce rhythms has been a well-known research subject. Many research results have been presented for modelling the human ability to recognize the rhythmic patterns from signals [Eck et al. 2000; Essens and Povel 1985; Jones and Boltz 1989; Large and Kolen 1994; Scarborough et al. 1992]. Those models were based on the analysis of the captured salient features of signals. In particular, Jones and Boltz observed that distinctive directional changes occur periodically in limb motions [Jones and Boltz 1989]. This observation provides us with a clue to capturing the rhythm of human motion. In radar, a similar problem has been studied to improve the performances of radar systems [Fogel and Gavish 1988; Sadler and Casey 1998]. Fogel and Gavish [Fogel and Gavish 1988] estimated the periodic interval from a given pulse signal with noise by performing spectral analysis. Sadler and Casey [Sadler and Casey 1998] formulated it as a linear regression problem. Our scheme for motion beat analysis is inspired by those results.

There have been quite a few works related to motion synthesis. For our purpose, we focus on the two categories of schemes for example-based motion synthesis: the schemes based on motion blending [Guo and Robergé 1996; Park et al. 2002; Rose et al. 1998; Wiley and Hahn 1997] and those based on posture rearrangement [Arikan and Forsyth 2002; Galata et al. 2001; Kovar et al. 2002; Lee et al. 2002; Li et al. 2002; Pullen and Bregler 2002; Tanco and Hilton 2000]. The main difference between the two categories is kinds of example motions to be used. The example motions for the former are labeled, and those for the latter are unlabeled.

The former category includes various motion blending schemes. Guo and Rovergé [Guo and Robergé 1996] and Wiley and Hahn [Wiley and Hahn 1997] provided interpolation techniques for example motions located regularly in parameter spaces. Rose *et al.* [Rose et al. 1998] and Sloan *et al.* [Sloan et al. 2001] introduced frameworks of motion blending based on scattered data interpolation with radial basis functions. Park *et al.* [Park et al. 2002] presented an on-line motion blending scheme to control virtual characters in real-time. To avoid singularities in motion blending, they represented joint orientations by unit quaternions. Provided with the temporal correspondence among the example motions, the motion blending schemes in this category can generate a desired motion in real time that satisfies the given constraints on both motion characteristics and environment geometry.

The latter category of schemes seamlessly rearrange postures in the example motions to obtain realistic motions [Arikan and Forsyth 2002; Galata et al. 2001; Kovar et al. 2002; Lee et al. 2002; Li et al. 2002; Pullen and Bregler 2002; Tanco and Hilton 2000]. Kovar *et al.* [Kovar et al. 2002] introduced a motion graph to represent the transitions among the poses of captured motion data. They also manually attached descriptive labels to motion clips for high-level control. Lee *et al.* [Lee et al. 2002] also represented captured motion data using a similar graph structure, and provided effective user interfaces for interactive character control. Arikan and Forsyth [Arikan and Forsyth 2002] applied a randomized algorithm for searching motions from a hierarchy of transition graphs. Pullen and Bregler [Pullen and Bregler 2002] developed a method for enhancing roughly-keyframed animation with captured motion data. In general, the posture rearrangement schemes can generate realistic motions while preserving details of the original motions. However, it may be time-consuming to search for desired motions from the graph when the number of example motions is large.

To exploit the advantages of both categories, we combine the idea of motion blending and that of posture rearrangement. After extracting all basic movements and their transitions from the unlabeled example motions based on motion beat analysis, our scheme classifies them into groups and constructs a movement transition graph whose nodes and edges respectively represent the prototype basic movements (groups) and their transitions reflecting the example motions. Given rhythmic sound together with kinematic constraints, our scheme traverses the graph while blending the labeled basic movements at each node in an on-line manner to synthesize a desired rhythmic motion.

3 Motion Beat Analysis

In this section, we present a novel scheme for extracting motion beats from given rhythmic motion data. Our scheme is based on the observation that distinctive directional changes periodically occur in real motions [Jones and Boltz 1989]. However, not every moment of rapid directional change necessarily corresponds to an actual motion beat but is just a candidate for a motion beat. Furthermore, we need to analyze the motion signals for all joints while compensating for their phase differences due to the hierarchical structure of a human body. We first describe how to find the most dominant period of the candidates obtained from the motion signal of a single joint. Then, we generalize this idea to simultaneously handle multiple joints. Using the dominant period, our scheme marks a sequence of moments called reference beats. Based on the reference beats, the actual motion beats are statistically estimated from the candidates obtained from all body parts.

3.1 Candidate Beat Extraction

Our motion data consist of a bundle of motion signals. Those signals are sampled at a sequence of discrete time instances with a uniform interval to form the corresponding sequence of frames. The sampled values from the signals at each frame determine the configuration of an articulated figure at that frame. We denote a motion (discrete motion signal) by $\mathbf{m}(i) = (\mathbf{p}(i), \mathbf{q}_1(i), ..., \mathbf{q}_J(i))^T$, where $\mathbf{p}(i) \in \mathbb{R}^3$ and $\mathbf{q}_1(i) \in \mathbb{S}^3$ describe the position and orientation of the root segment, $\mathbf{q}_j(i) \in \mathbb{S}^3$ gives the orientation of joint j at frame i, and J is the number of joints. For convenience, the root is often referred to as a joint ($j = 1$).

To identify the candidates for motion beats, we extract the moments of rapid directional changes in a motion. Such moments can be detected from the zero-crossings of the second derivative of each

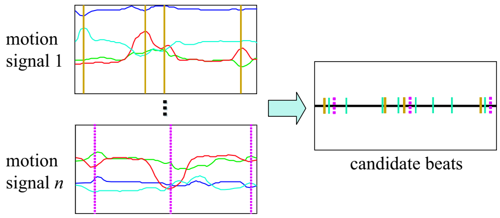

Figure 1: Phase differences among motion signals.

motion signal at every frame, which can be computed from the linear or angular acceleration of the signal. We concentrate on extracting the zero-crossing moments from orientation signals. For position data such as the root and end effector positions, we refer the readers to previous results [Bindiganavale and Badler 1998]. For beat analysis, we can use either joint positions or joint orientations. In our experiments, both of those data have been equally good for extracting motion beats. However, we prefer joint orientations since they are intrinsic parameters and facilitate more convenient posture control of articulated figures.

The angular velocity of an orientation curve $\mathbf{q}(t)$ in a continuous domain is $\omega(t) = 2\mathbf{q}^{-1}(t)\dot{\mathbf{q}}(t)$. However, we cannot apply this differential equation to a discrete motion signal $\mathbf{q}_j(i)$ since the exact value of $\dot{\mathbf{q}}_j(i)$ is not available. Instead, we approximate the angular velocity $\omega_j(i)$ of joint j using the angular velocity of the *slerp* (spherical linear interpolation) motion [Shoemake 1985] that starts at $\mathbf{q}_j(i-1)$ toward $\mathbf{q}_j(i)$. That is,

$$\omega_j(i) = \frac{2\log(\mathbf{q}_j^{-1}(i-1)\mathbf{q}_j(i))}{h} \in \mathbb{R}^3, \tag{1}$$

where h is the time interval between frames $i-1$ and i. It can be proved that the approximate angular velocity in Equation (1) converges to the exact angular velocity, $\omega(t) = 2\mathbf{q}^{-1}(t)\dot{\mathbf{q}}(t)$, as the time interval h approaches zero [Kim 1996]. Thus, we approximate the angular acceleration $\alpha_j(i)$ of joint j at frame i as follows:

$$\alpha_j(i) \approx \frac{\omega_j(i) - \omega_j(i-1)}{h}. \tag{2}$$

We choose all zero-crossing moments of the motion signal as the candidate motion beats, which can be obtained by detecting the zero-crossing of every component of $\alpha_j(i)$.

The sequence of candidates extracted from a motion signal embeds a periodic pattern with some missing observations, which can possibly be contaminated with outliers. As shown in Figure 1, the candidate beats have a complicated distribution due to the phase differences among the movements of body parts. Therefore, the outliers are hard to be distinguished from the actual motion beats.

3.2 Reference Beat Estimation

We first provide a method for finding the dominant period for the candidates obtained from a single motion signal. Then, we generalize this scheme for a bundle of motion signals to estimate the reference beats.

Single joint: A problem of estimating the dominant period of a candidate sequence can be transformed to that of estimating the pulse repetition interval (PRI) from periodic pulse trains based on their arrival times only [Sadler and Casey 1998].

Let S be the sequence of candidate beats, that is, $S = \{t_j\}_{j=1}^N$. Each t_j, $1 \le j \le N$, is a moment of zero-crossing, which can be modeled as follows:

$$t_j = \phi + k_j T + \eta_j, \quad j = 1, ..., N \tag{3}$$

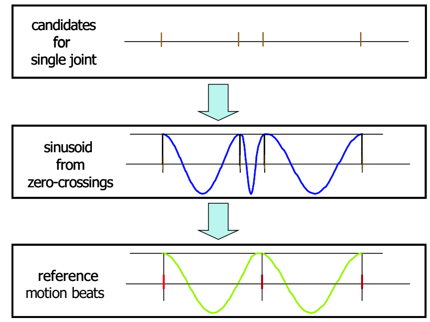

Figure 2: Estimating the reference beats from a motion signal.

where T is the unknown period of beats, ϕ is a random phase with a uniform distribution over an interval $[0, T)$, that is, $\phi \sim \mathcal{U}[0, T)$, k_j, $1 \le j \le N$ are positive non-repeating integers, and η_j, $1 \le j \le N$ are zero-mean additive white Gaussian noises in the interval $[-T/2, T/2)$.

As previously described, not every moment of zero-crossing is necessarily a motion beat. For example, some of t_j, $1 \le j \le N$ can possibly be outliers with a uniform distribution. For those outliers, their moments of zero-crossing do not correspond to noisy integer multiples of T as given in Equation (3). Instead, they can be completely random in time. Without loss of generality, we assume that $t_j < t_{j+1}$ for $j = 1, ..., N-1$.

To obtain the dominant period of S, we estimate the dominant frequency of a sinusoidal function that has its upper extreme value at t_j for $1 \le j \le N$, that is, a sinusoidal function $y(t)$ such that $y(t_j) = 1$ for all j as shown in Figure 2. For simplicity in sinusoid fitting, we construct $(N-1)$ cosine curves each of which spans a time interval $[t_{k-1}, t_k)$ with a single cycle for all k, that is, the sinusoidal function $y(t)$ is defined as

$$y(t) = \begin{cases} \cos(2\pi(\frac{t-t_{k-1}}{t_k-t_{k-1}})) & \text{if } t_{k-1} \le t < t_k,\ k = 2, ..., N \\ 1 & \text{if } t = t_N. \end{cases} \tag{4}$$

$y(t)$ is not a sinusoidal function in a strict sense, since it does not guarantee even C^1 continuity at the joining points, $(t_j, y(t_j))$. However, their deviations from a perfect sinusoidal function are small enough to be regarded as noise.

We assume that the beat period is not less than the inter-frame time of the motion, that is, the highest possible frequency component of $y(t)$ is a reciprocal of the inter-frame time. Based on the *Nyquist sampling theorem* [Proakis and Manolakis 1996], we regularly sample M values from $y(t)$, that is, $\tilde{t}_j$ for $j = 1, ..., M$, with a sampling rate of more than twice the highest possible frequency. Then, we perform spectral analysis using the periodogram:

$$P_y(f) = \frac{1}{M}|\sum_{j=1}^{M} y(\tilde{t}_j)e^{2\pi i f \tilde{t}_j}|^2. \tag{5}$$

The power spectrum density (PSD) of $y(t)$ has a distinctive peak at the dominant frequency [Gomes and Velho 1999; Proakis and Manolakis 1996]. As illustrated in Figure 2, the most dominant frequency of the candidate beats is the frequency $\tilde{f}$ that gives the peak value for $P_y(\tilde{f})$:

$$\tilde{f} = \arg\max_f P_y(f), \tag{6}$$

from which we obtain the dominant period $\tilde{T} = \frac{1}{\tilde{f}}$. Then, we esti-

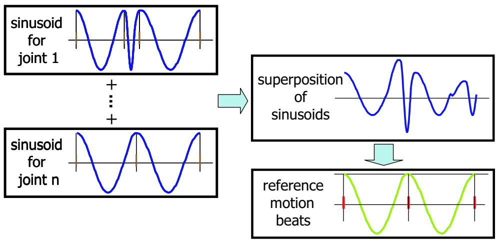

Figure 3: Estimating the reference beats from a bundle of motion signals.

mate the phase ϕ as follows:

$$\tilde{\phi} = \arg\max_{\phi} \sum_{j=1}^{M} y(\tilde{t}_j) y(\tilde{t}_j + \phi), \quad 0 \le \phi < \tilde{T}. \qquad (7)$$

The k-th reference beat, r_k, $k = 1, ..., N$ is finally computed by $r_k = \tilde{\phi} + k\tilde{T}$, which would be coincident with the corresponding motion beat for a perfectly-rhythmic motion.

Multiple joints: The period estimation scheme for a single motion signal can be extended to multiple motion signals. As illustrated in Figure 3, we first superpose (add) K sinusoidal functions $y_1(t), ..., y_K(t)$ constructed from the corresponding candidate beat sequences, where K is the number of joints for an articulated figure. Then, we compute the PSD of the superposed signal $S(t) = y_1(t) + ... + y_K(t)$:

$$P_S(f) = \frac{1}{M} |\sum_{j=1}^{M} S(\hat{t}_j) e^{2\pi i f \hat{t}_j}|^2, \qquad (8)$$

where $\hat{t}_j$, $1 \le j \le M$ are defined similarly as in Equation (5). The dominant period $\hat{T}$ and the phase $\hat{\phi}$ of the composite sequence $S(t)$ can respectively be estimated from $P_S(f)$ and $S(\hat{t}_j)$ as described in the case of a single joint, to determine the reference beat r_k, $1 \le k \le N$.

3.3 Motion Beat Estimation

We now estimate the actual motion beats from the candidates guided by the reference beats. The actual beats are not perfectly coincident with the reference beats unlike in an ideal case. However, most candidate beats except some outliers are likely to be clustered around the reference motion beats as depicted in Figure 4. The representative of each cluster is chosen as the actual beat using a Huber M-estimator [Huber 1981]. For each reference beat r_k, we estimate the representative $\hat{b}_k$ from the candidates within the window $[r_k - \hat{T}/2, r_k + \hat{T}/2]$ as follows:

$$\hat{b}_k = \arg\min_{b} \sum_{j=0}^{M} W_k(\tilde{t}_j) \rho(\tilde{t}_j - b), \qquad (9)$$

where $\rho(x)$ is a Huber function:

$$\rho(x) = \begin{cases} x^2/2 & \text{if } |x| \le \alpha \\ k(|x| - k/2) & \text{if } |x| > \alpha, \end{cases} \qquad (10)$$

for a tuning constant α, and $W(x)$ is a window function:

$$W_k(x) = \begin{cases} 1 & \text{if } x \in [r_k - \hat{T}/2, r_k + \hat{T}/2] \\ 0 & \text{otherwise.} \end{cases} \qquad (11)$$

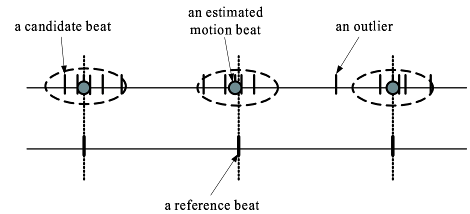

Figure 4: Clusters of candidate beats.

$\hat{b}_k$ is computed numerically with the downhill simplex minimization algorithm [Press et al. 1999]. The reference beat r_k is used as an initial guess for the solver.

Motion signals may miss distinctive directional changes at some motion beats, that is, it is possible that no candidate beats are detected in a range $[r_k - \hat{T}/2, r_k + \hat{T}/2]$ for some k. In this case, we designate the reference beat r_k itself as a motion beat $\hat{b}_k$.

4 Movement Transition Graph

In this section, we show how to construct a movement transition graph of which the nodes and edges represent the prototype movements and their transitions, respectively. Given basic movements extracted from the unlabeled example motions, we classify those movements into groups based on their similarities such that the basic movements in a group can be thought of as variants of the same labeled basic movement called a prototype movement. Then, we estimate the likelihoods of transitions between the groups of basic movements by analyzing the example motions.

4.1 Basic Movement Classification

Let $\mathcal{E} = \{\mathbf{M}_1, \mathbf{M}_2, ..., \mathbf{M}_N\}$ be a set of example motions. Each example motion $\mathbf{M}_i$, $1 \le i \le N$ consists of a sequence of basic movements, that is, $\mathbf{M}_i = (\mathrm{m}_i^1\ \mathrm{m}_i^2\ ...\ \mathrm{m}_i^{N_i})$, where m_i^c, $1 \le c \le N_i$, and N_i are a basic movement and the number of basic movements in $\mathbf{M}_i$, respectively.

We assume that the type of every example motion $\mathbf{M}_i$, $1 \le i \le N$ such as the *Waltz* and the *Tango* has been already specified at the motion-capture stage. The motion type uniquely determines the rhythmic pattern, that is, a sequence of motion beats. We decompose the example motion $\mathbf{M}_i$ into basic movements m_i^c, $1 \le c \le N_i$ such that the motion beats of every movement m_i^c are coincident with the rhythmic pattern of $\mathbf{M}_i$ called a *meter*. Since the example motions may have different tempos, the inter-beat intervals called *beat periods* of basic movements may also be different from each other. To make their beat periods identical, we align all basic movements by timewarping, interpreting the motion beats as keytimes.

Even apparently-similar basic movements may have quite different root positions and orientations, since the configurations of root segments are represented in a global coordinate system. To measure the difference between the basic movements in a coordinate system compatible to all of them, we make all basic movements be located at a fixed position while facing a fixed direction as described in [Kovar et al. 2002]: We translate them along the horizontal (zx) plane and rotate them about the vertical (y) axis so that their root segments at the first frame are on the y-axis and face the $+z$ direction.

Let $\mathcal{S}_k$, $1 \le k \le K$ be the groups of similar basic movements for the example motions in $\mathcal{E}$. Those groups are obtained by clustering the basic movements extracted from the example motions. For clustering, we need a difference measure between basic movements. Consider a pair of basic movements, $\mathrm{m}_1(i) =$

$(\mathbf{p}^1(i), \mathbf{q}_1^1(i), ..., \mathbf{q}_J^1(i))$ and $\mathrm{m}_2(i) = (\mathbf{p}^2(i), \mathbf{q}_1^2(i), ..., \mathbf{q}_J^2(i))$, where $\mathbf{p}^c(i)$ and $\mathbf{q}_j^c(i)$ for $1 \leq j \leq J$ and $c = 1, 2$ are the position of the root segment and the orientation of joint j at frame i of movement m_c as defined in Section 3.1. Motivated by a result of copyright protection for choreographic works [J. van Camp 1994], we derive a new difference measure from the posture correspondence at every motion beat.

At frame i, we define the difference measure D between movements, m_1 and m_2 as suggested in Lee *et al.* [Lee et al. 2002]:

$$\mathrm{D}(\mathrm{m}_1(i), \mathrm{m}_2(i)) = d_a(\mathrm{m}_1(i), \mathrm{m}_2(i)) + \nu d_v(\mathrm{m}_1(i), \mathrm{m}_2(i)). \quad (12)$$

The first and second terms respectively represent the posture difference and the velocity difference, and ν is a weight value for the second term. The first term reflects the differences in the root position and joint angles:

$$d_a(\mathrm{m}_1(i), \mathrm{m}_2(i)) = \|\mathbf{p}^1(i) - \mathbf{p}^2(i)\|^2 + \sum_{j=1}^{J} w_j \|\log(\mathbf{q}_j^1(i)^{-1}\mathbf{q}_j^2(i))\|^2, \quad (13)$$

where w_j, $1 \leq j \leq n$ is the weight value. The velocity term is for comparing the dynamics of motions and is represented as the sum of the weighted velocity differences:

$$d_v(\mathrm{m}_1(i), \mathrm{m}_2(i)) = \|\mathbf{v}^1(i) - \mathbf{v}^2(i)\|^2 + \sum_{j=1}^{J} v_j \|\omega_j^1(i) - \omega_j^2(i)\|^2, \quad (14)$$

where $\mathbf{v}^c(i)$, $c = 1, 2$, is the linear velocity of the root segment of movement m_c at frame i, $\omega_j^c(i)$, $1 \leq j \leq n$ is the angular velocity of joint j, and v_j is the weight value.

Now, the difference between two movements, m_1 and m_2 is measured by the sum of posture differences at their corresponding motion beats, $\{b_1, ..., b_B\}$:

$$\mathcal{D}(\mathrm{m}_1, \mathrm{m}_2) = \sum_{i=1}^{B} \mathrm{D}(\mathrm{m}_1(b_i), \mathrm{m}_2(b_i)), \quad (15)$$

where b_i is the i-th motion beat of m_1 and m_2, and B is the number of motion beats. Based on this difference measure, we can classify all basic movements into groups. There are two cases depending on the availability of the number of groups.

Suppose that the number K of groups is given in advance. Let $\mathbf{S} = \{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_K\}$ denote the set of all those groups. Then, we can employ a well-known *K-means clustering method* [Bezdek 1981; Dubes and Jain 1976; Forgy 1965; Lee et al. 2002] to identify them. Let $\mathcal{X}$ be a set of the basic movements obtained from $\mathcal{E}$, that is, $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ where $\mathbf{x}_i$, $1 \leq i \leq n$ and n denote a basic movement and the total number of basic movements, respectively. The K-means clustering method partitions $\mathcal{X}$ into groups $\mathcal{S}_k$, $1 \leq k \leq K$ by minimizing the following objective function $f(\mathbf{S}, \mathcal{X})$:

$$f(\mathbf{S}, \mathcal{X}) = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathcal{S}_k} \mathcal{D}(\mathbf{x}_i, \mathbf{c}_k), \quad (16)$$

where $\mathbf{c}_k$ is the group center for $\mathcal{S}_k$. We optimize the objective function by iterative estimations of $\mathbf{c}_k$ and $\mathcal{S}_k$. Initially, the movements are assigned at random to K groups. At each iteration, every $\mathbf{x}_i$ is assigned to a group $\mathcal{S}_k$ such that its difference from the group center $\mathbf{c}_k$ is minimized. Then, the group centers are reestimated from $\mathcal{S}_k$ as follows:

$$\mathbf{c}_k = \arg\min_{\mathbf{x} \in \mathcal{S}_k} \sum_{\mathbf{x}_i \in \mathcal{S}_k} \mathcal{D}(\mathbf{x}, \mathbf{x}_i), \quad 1 \leq k \leq K. \quad (17)$$

These two steps are alternated until there is no further change in the assignment. This process may be trapped in a local minimum. Therefore, we may try the process with different initial assignments for better clustering.

Now, suppose that the information on the number K of groups is not available in advance. Then, we use the *subtractive clustering method* [Chiu 1994], which is a fast one-pass algorithm for estimating the number of groups in a data set and the group centers. The subtractive clustering method assumes that each $\mathbf{x}_i$ has a potential to be a group center. The potential P_i of $\mathbf{x}_i$ is given by

$$\mathrm{P}_i = \sum_{j=1}^{n} \exp(-\frac{\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j)}{r^2}), \quad i = 1, ..., n, \quad (18)$$

where r is a parameter indicating the range of influence of a movement over other movements. We choose the movement $\mathbf{x}_m$ with the highest potential P_m as the first group center, and then remove the influence of the potential P_m from the others as follows:

$$\mathrm{P}_i' = \mathrm{P}_i - \mathrm{P}_m \exp(-\frac{\mathcal{D}(\mathbf{x}_i, \mathbf{x}_m)}{sr^2}), \quad i = 1, ..., n, \quad (19)$$

where $s > 1$ is a constant. The constant s controls the range of $\mathbf{x}_m$ within which the existence of other group centers are discouraged. We select the basic movement with the highest remaining potential as the next group center and remove its influence from the remainders. This process is repeated until the potentials of all remaining movements fall below a threshold. Based on the estimated number of groups and their centers, every $\mathbf{x}_i$ is assigned to a group $\mathcal{S}_k$ of which the center has the greatest influence over it, that is, $\mathcal{S}_k$ such that

$$k = \arg\max_{j \in [1,K]} \exp(-\frac{\mathcal{D}(\mathbf{x}_i, \mathbf{c}_j)}{r^2}) \quad (20)$$

where $\mathbf{c}_j$ is the center of $\mathcal{S}_j$.

4.2 Graph Construction

With the groups of basic movements identified, we now describe how to construct a movement transition graph by finding the transitions among the prototype movements representing the groups. Those transitions are established based on two factors, *kinematic continuity* and *behavioral continuity*: The former measures the smoothness of a transition, and the latter reflects the degree of satisfying the transition rules such as choreographic patterns observed in the example motions.

Kinematic continuity: The kinematic continuity has been widely used for modeling transitions between motion segments or motion frames [Arikan and Forsyth 2002; Kovar et al. 2002; Lee et al. 2002]. The likelihood of the transition from a basic movement to a basic movement (possibly itself) can be estimated using the difference between the last posture of the former and the first posture of the latter. We define the difference measure between the groups as

$$D(\mathcal{S}_i, \mathcal{S}_j) = \sqrt{\frac{\sum_{\mathrm{m}_e \in \mathcal{S}_i, \mathrm{m}_s \in \mathcal{S}_j} \mathrm{D}(\mathrm{m}_e(L), \mathrm{m}_s(F))^2}{|\mathcal{S}_i||\mathcal{S}_j|}}, \quad (21)$$

where F and L respectively denote the first frame of m_e and the last frame of m_s, and $|\mathcal{S}_i|$ and $|\mathcal{S}_j|$ are respectively the number of basic movements in $\mathcal{S}_i$ and that in $\mathcal{S}_j$. Then, we can determine the likelihood $\mathcal{P}_k(i, j)$ of the transition from $\mathcal{S}_i$ to $\mathcal{S}_j$ as follows:

$$\mathcal{P}_k(i, j) = w_i \exp(-D(\mathcal{S}_i, \mathcal{S}_j)/\sigma), \quad (22)$$

where σ is a control parameter specified by a user. The weight value w_i is determined so that $\sum_{j=1}^{K} \mathcal{P}_k(i, j) = 1$ for all i, where K is the number of the movement groups.

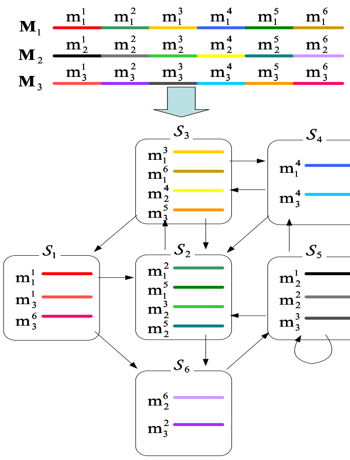

Figure 5: A set of example motions and their movement transition graph.

Behavioral continuity: The behavioral continuity captures the natural transitions among the prototype movements of the example motions. From the movement transitions between the groups observed in the example motions, the likelihoods of the transitions between the corresponding prototype movements are estimated to represent the behaviors of example motions. Figure 5 illustrates a set of example motions and their movement transition graph, where similar basic movements are marked with similar colors and classified into the same group.

Let $\mathcal{P}_b(i,j)$ and $\pi(i)$ respectively be the likelihood of the transition from $\mathcal{S}_i$ to $\mathcal{S}_j$ and the probability that a motion starts from $\mathcal{S}_i$:

$$\begin{aligned} \mathcal{P}_b(i,j) &= P[\mathbf{s}_{t+1} = \mathcal{S}_j | \mathbf{s}_t = \mathcal{S}_i], & (23)\\ \pi(i) &= P[\mathbf{s}_1 = \mathcal{S}_i], & (24) \end{aligned}$$

where $\mathbf{s}_t$ is the state of the transition model at time t. To learn the transition model automatically, we employ Baum-Welch's EM (expectation-modification) method [Li and Parizeau 2000; Rabiner 1989]. This method iteratively reestimates the probabilities, $\mathcal{P}_b(i,j)$ and $\pi(i)$ as follows:

$$\overline{\mathcal{P}}_b(i,j) = \frac{\text{expected number of transitions from } \mathcal{S}_i \text{ to } \mathcal{S}_j}{\text{expected number of transitions from } \mathcal{S}_i}, \quad (25)$$

$$\overline{\pi}(i) = \frac{\text{expected number of times in } \mathcal{S}_i \text{ at time } t = 1}{\sum_{k=1}^{K} \text{expected number of times in } \mathcal{S}_k \text{ at time } t = 1}. \quad (26)$$

These probabilities reflect the behavioral continuity as observed in the example motions. The initial state distribution $\pi(i)$ may be specified manually when sufficient example motions are not available.

With the groups of basic movements and the transitions among the groups identified, we are ready to describe how to construct the movement transition graph. Each node $\mathbf{n}_i$, $1 \le i \le K$ represents a group $\mathcal{S}_i$ of basic movements. The likelihood $\mathcal{P}(i,j)$ of the transition from a group $\mathcal{S}_i$ to a group $\mathcal{S}_j$ can be expressed as the weighted sum of kinematic and behavioral continuities as follows:

$$\mathcal{P}(i,j) = w_1 \mathcal{P}_k(i,j) + w_2 \mathcal{P}_b(i,j), \quad (27)$$

where $\mathcal{P}_k(i,j)$ and $\mathcal{P}_b(i,j)$ are likelihoods defined in Equations (22) and (25), respectively, and w_1 and w_2 are weight values such that $w_1 + w_2 = 1$, $w_1 \ge 0$, and $w_2 \ge 0$. A node $\mathbf{n}_i$ is connected to a node $\mathbf{n}_j$ with an edge if $\mathcal{P}(i,j)$ is larger than a given threshold. $\mathcal{P}(i,j)$ is renormalized after identifying the edges such that $\sum_j \mathcal{P}(i,j) = 1$ for all node i. The resulting movement transition graph represents the transitions among basic movements reflecting the example motions as illustrated in Figure 5.

4.3 Refinement

After constructing the movement transition graph, we may edit this graph interactively. For example, ballroom dances such as the *Waltz* and the *Tango* have choreographic rules, that is, transition rules that specify preferable movements followed by each basic movement [Silvester 1993]. The likelihoods of transitions learned from the example motions may not perfectly reflect the choreographic rules when the example motions are not sufficient enough to reflect the rules properly. Thus, we may connect nodes of which transitions are left out and adjust the likelihoods of transitions in accordance with the choreographic rules. In particular, we avoid dead-ends, that is, nodes with no outgoing edges by interactively connecting them with another nodes. We may also remove those nodes automatically as described in [Kovar et al. 2002; Lee et al. 2002; Schödl et al. 2000].

5 Motion Synthesis

Given a rhythmic sound signal such as music together with the direction and speed to move, our objective is to generate a new rhythmic motion on the fly. The direction and speed are either given explicitly *or* derived from the environment changing dynamically. The resulting motion should satisfy both temporal and spatial constraints prescribed by the characteristics of a given sound and the motion specification. We assume that the input sound signal is specified in a MIDI format so as to provide the constituent music beats directly. We also specify interactively the type of sound signal such as the *Waltz* and the *Tango* once at the beginning.

For motion synthesis, we prepare a library of movement transition graphs each of which represents a collection of rhythmic motions of an identical type. From the library, a movement transition graph is chosen so that the type of motions represented by the graph matches that of the input sound signal. Starting from a node $\mathcal{S}_i$ chosen at random according to its probability $\pi(i)$ of being the initial basic movement, we traverse the movement transition graph from node to node, guided by the transition probabilities until the sound signal ends, while synthesizing a basic movement at each node. By the way in which we have connected a pair of nodes by an edge, a basic movement synthesized at a node can be stitched with that at the next node seamlessly, since the final posture of the former is sufficiently similar to the starting posture of the latter.

We employ the on-line motion blending scheme as given in [Park et al. 2002] to generate a basic movement at a node. Based on multidimensional scattered data interpolation, this scheme consists of four steps: parameterization, weight computation, time-warping, and posture blending. In the pre-processing step, all basic movements are parameterized according to their characteristics such as speed and turning angle. Provided with the parameters of a target motion such as motion type, direction, and speed extracted from the motion specification, the next step is for computing the contribution of every basic movement to the target movement using cardinal basis functions [Sloan et al. 2001]. The third step is for incremental time-warping. We adapt this step for synchronizing the basic movements with the input sound signal on the fly. Specifically, the motion beats of every basic movement are aligned with the music beats. The final step is for blending the time-warped movements in accordance with their contributions to synthesize the target movement.

The size and proportions of a target character may be different from those of the performer for the captured motions. Therefore, we cannot apply the blended motion directly to the target character. Otherwise, artifacts such as foot sliding or penetration would appear in the resulting images. To avoid those artifacts, the target stance foot position is first computed at each frame by blending the stance foot positions of example motions. Then, we employ the real-time motion retargeting algorithm in [Shin et al. 2001], which adapts the motion to the target character and the environment.

6 Experimental Results

We perform experiments on an Intel® Pentium® PC (P4 2.2GHz processor and 1GB memory). Our human model had 43 DOFs (Degrees Of Freedom) that consist of 6 DOFs for the pelvis, 3 DOFs for the spine, 7 DOFs for each limb, 3 DOFs for the neck, and 3 DOFs for the head. The motion clips were captured at the rate of 120 frames per second and then down-sampled at the rate of 30 frames per second for real-time display. Input sound signals were given in the type 0 standard MIDI format. We first demonstrate experimental results for motion beat analysis, and then show those for rhythmic motion synthesis.

6.1 Motion Beat Analysis

To evaluate the accuracy of our motion beat analysis scheme, experiments were performed for two kinds of motions: motions with regular motion beats and those with random motion beats. For each of these cases, the differences of the estimated beats from their corresponding actual beats were within one frame.

Make-up Motions: We created a motion with regular motion beats by repeatedly concatenating a short cyclic walking motion. Since we knew the moments of the distinctive directional change and the length of the motion, the exact motion beats could be obtained. The length and beat period were 900 frames and 15 frames, respectively.

A motion with irregular beats was similarly obtained by perturbing the moments of distinctive directional change of the cyclic motion randomly within 15% of its length. With the magnitude of each perturbation known, we could mark the actual motion beats exactly by hand. The length of the test motion was 900 frames.

Figures 6(a) and (b) show the errors between the estimated beats and the actual beats for the motion with the regular beats and those for the motion with the irregular beats, respectively. For the motion with the regular beats, no errors were observed as expected. For the motion with the irregular beats, the errors were distributed within one frame, and their mean and variance were 0.3233 and 0.2999, respectively.

Dance Motions: We applied our scheme to a real rhythmic motion, that is, a captured dance motion. The tempo of background music was 150 beats per minute. Since the actual motion beats were not known in advance, we marked them manually by referring to the beats of the background music acquired together with the dance motion. The length and beat period of the test motion were 714 frames and 12 frames, respectively. The differences of the estimated beats from the corresponding manually-marked beats were within one frame as shown in Figure 6-(c). Their mean and variance were 0.3787 and 0.1344, respectively.

6.2 Rhythmic Motion Synthesis

We demonstrated the capability of our scheme for rhythmic motion synthesis through five experiments. In all experiments, our scheme generated motions extremely fast. For the first three experiments, a request for motion generation of a character can be handled at more

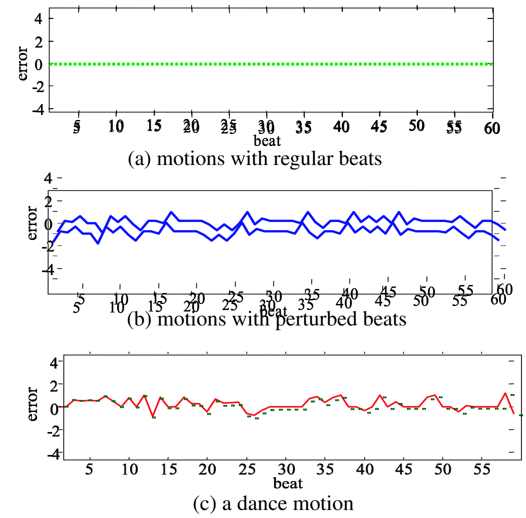

Figure 6: The accuracy of the estimated motion beat: The horizontal and vertical axes represent the beat number and the error at each beat, respectively.

than 2000 Hz. The fourth experiment is most time-consuming, for which frame rate is 46 Hz. The frame rate of the last experiment is 50 Hz.

Synchronization: Our first experiment was for showing the difference of a synchronized motion from an unsynchronized motion. We created an unsynchronized test motion by simply stitching the blended basic movements without timewarping while traversing the movement transition graph in accordance with the likelihoods of transitions. Those movements were obtained from two example motions of different tempos as will be explained later. The motion synchronized with background music was obtained using our scheme while following the same path in the movement transition graph. The characters on the right and left hand sides in Figure 7 show synchronized and unsynchronized motions, respectively. Their difference can be observed more clearly in the accompanying movie clip.

We used two free-style modern dances of the $4/4$ meter as example motions. The beat periods of the corresponding motion clips were 12 and 13 frames, respectively, and their total length was 2367 frames. Our scheme decomposed the example motions into 48 basic movements by punctuating them at every four beats, and then classified the basic movements to construct a movement transition graph. The resulting movement transition graph was composed of 37 nodes. On average, 1.3 basic movements were distributed to each node. For synchronization, the background music was given in a MIDI track.

Interactive Choreography: To show the on-line, real-time capability of our motion synthesis scheme, we interactively choreographed a new motion synchronized with a given piece of music. We moved the mouse pointer in an on-line manner to make the target character dance to the music while chasing the pointer as illustrated in Figure 8. During motion synthesis, motion parameters such as speed and direction were computed from the sampled pointer position and the current posture of the target character.

Our example motions were modern dances, which do not have standardized sets of basic movements unlike other forms of dances such as the *ballet or* the *Waltz*. However, owing to the property of dances, that is, the repetition of similar movements with variations [Hawkins 1988; Minton 1997], we can construct the move-

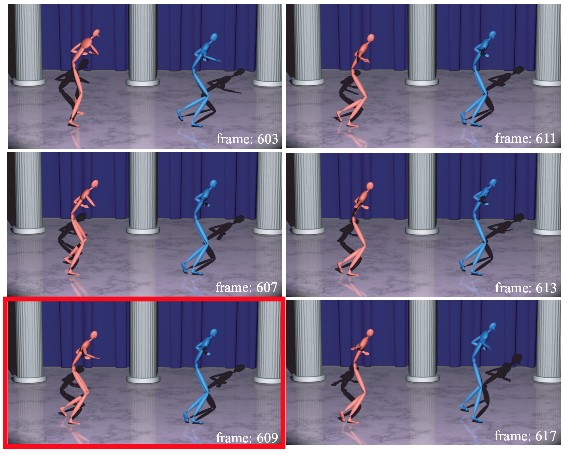

Figure 7: Synchronized vs. unsynchronized motions: Frame 609 (surrounded by the red rectangle) corresponds to the beat.

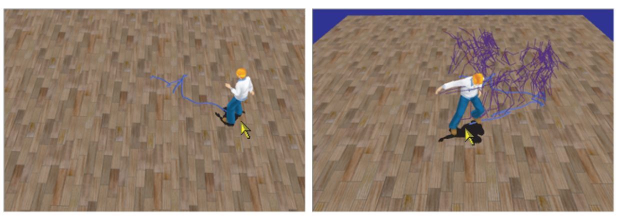

Figure 8: Interactive choreography.

ment transition graph using a small number of example motions. We captured the example motions from a performer dancing to a piece of background music of the $4/4$ meter. Thus, basic movements are obtained by punctuating the example motions at every four beats. We used three captured example motions and their mirror-reflected motions, of which the total length and beat period were 5100 and 12 frames, respectively. The movement transition graph was composed of 34 nodes which were constructed from 108 basic movements. 3.2 basic movements were assigned to each node on average.

Waltzing Couple: We constructed a movement transition graph for a coupled motion (see Figure 9). Assuming that the motions of a couple have a common rhythmic structure, we obtained the motion beats by analyzing their motion signals simultaneously. Each node of the constructed movement transition graph represents variations of a prototype movement of the couple observed in the example motions. The total length and beat periods of example motions were 1829 and 20 frames, respectively. 30 basic movements of the *Waltz* were captured from the motions. The constructed movement transition graph had five nodes each of which had six coupled movements on average.

The *Waltz* is composed of basic movements for flowing zigzag around a ballroom [Bottomer 1998; Silvester 1993]. Every basic movement has three beats each of which respectively corresponds to *forward* or *backward*, *left-side* or *right-side*, and *close* steps. Figure 9 (left) shows a simple Waltz figure adopted for our experiment. The movements containing a *backward* step are rarely used for moving forward. Similarly, those containing a *forward* step are not used in general for moving backward. To form zigzagged moving patterns, the movements containing a *left-side* step is more likely to be followed by those containing a *right-side* step than others, and vice versa. Example motions included a variety of basic movements mentioned above. Our scheme can handle any kind of Waltz figures as long as their motion beats match the corresponding music beats.

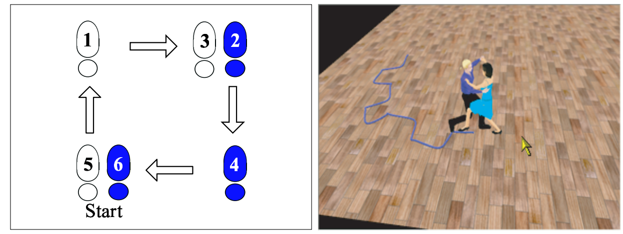

Figure 9: A waltzing couple. (Left) A basic *Waltz* step for man: the white and blue feet represent the left and right feet, respectively. (Right) Interactive control of the couple.

A novel coupled motion was synthesized based on those rules. Our scheme prescribed the parameters of a movement to be synthesized at a node guided by the mouse pointer, and selected the next node to transit in accordance with the pre-computed likelihoods of transitions attached to the edges incident from the node. To generate a motion for the coupled characters, the parameters of a desired basic movement for the leader (man) were first computed to obtain the weights for blending (see Section 5). Then, the movement of the couple was produced by blending their basic movements at the current node with those weights. The parameters included the speed, direction, and body orientation of the leader determined from the goal position specified by the mouse pointer and the posture of the character at the current frame.

Ballroom Dance: We applied our scheme to a crowd dancing in a ballroom. The example motions and their control parameters were the same as those of the previous experiment. Based on the work in [Helbing and Molnar 1995; Helbing et al. 2000; Kim et al. 2003], we determined the behaviors of couples by combining the *flow forces* for couples to travel around the room and the *interaction forces* for avoiding collisions with other couples *or* static objects in the room such as walls and columns. Since the natural flow of the *Waltz* is moving around the room [Bottomer 1998], we were able to automatically generate the flow force by creating a vortex. The interaction force exerted on a couple was adaptively determined to react to nearby approaching couples and static objects, which can be sensed by employing the event-driven collision detection algorithm in [Kim 2002; Kim et al. 2003]. The control parameters such as speed, direction, and body orientation were determined from the resulting force field. With this crowd behavior model, we produced an animation of 20 waltzing couples in real time (46 frames per second) as illustrated in Figure 10.

Marching Soldiers: We finally applied our scheme to locomotive motions to synthesize a scene with troops marching in step synchronized with a piece of military music. We captured marching motions of various speed and turning angle. These motions were 1299 frames long in total, and their beat periods were 16 frames. Since the period of the marching motion is two beats, we decomposed the example motions into 44 basic movements each consisting of two motion beats. The movement transition graph was composed of two nodes each of which contains 22 basic movements on average. In order to preserve the formation of the troops while turning left *or* right, we used different speed and turning angle for the characters at the inside corner than those at the outside corner (see Figure 11 (left)). To do this, we fitted an offset curve of the B-spline curve representing the street center for each row of the soldiers, and then determined the speed and turning angle from the tangent vector of the curve at each frame. We simulated the 40 marching soldiers in real time (50 frames per second).

Figure 10: A ballroom scene

Figure 11: Marching soldiers

7 Discussion

Unlike previous work based on frame-level transitions [Arikan and Forsyth 2002; Kovar et al. 2002; Lee et al. 2002], our motion synthesis scheme is based on transitions between basic movements. The frame-level transitions may not preserve the rhythms of motions, since transitions occur from frame to frame without considering beat patterns. Accordingly, distinctive directional changes at motion beats may be vanished in a synthesized motion.

Our beat analysis scheme estimates the motion beats automatically from a given rhythmic motion without any additional information such as background music. Motion beat analysis is required even if the background music is captured together with the rhythmic motion: Due to various sources of noise, the motion beats of the captured motion may not be coincident with the corresponding music beats. Simple rearrangement of noisy basic movements, obtained by punctuating the input motion only with the help of the music beats, may yield a motion of which the distinctive directional changes drift from the music beats.

We have determined the likelihoods of transitions among the basic movements based on the proposed measure reflecting both *kinematic continuity* and *behavioral continuity* (see Section 4). When sufficient example motions are available, we can construct the movement transition graph by placing more weight on the behavioral continuity so that the graph reflects the natural transitions among basic movements. Otherwise, by placing more weight on the kinematic continuity, the movement transition graph can generate a variety of motions from a small number of example motions.

In motion synthesis, we have parameterized basic movements by speed and turning angle. Those parameters are extracted from the trajectory of each movement by approximating it to an arc. For basic movements with complex trajectories such as those in waltzing, we approximate their global moving patterns to linear arcs (of infinite radius) by connecting the initial and final positions while regarding the differences between the trajectories and their corresponding arcs as motion details.

8 Conclusion

We have proposed a novel scheme for synthesizing a new rhythmic motion from unlabelled example motions. Our scheme extracts the motion beats from the example motions to capture the basic movements and their transitions, and then constructs a movement transition graph using those data. Given rhythmic sound together with kinematic constraints, our scheme traverses the graph from node to node guided by transition probabilities while blending the basic movements at each node, to synthesize a desired rhythmic motion. We exploit both advantages of motion blending and posture rearrangement. For efficiency and controllability of motion synthesis, we adopt the idea of motion blending. For preservation of rhythmic patterns, we generalize the idea of posture rearrangement to the level of basic movements.

Our scheme can be applied to automatically generating dancing and locomotive motions of characters, for example, a dancing crowd, a musical band performing a piece of music, and a troop of soldiers marching down the street. Our scheme can be used for synchronizing rhythmic motions with audio tracks, which has been a labor-intensive task in producing computer animations. Choreographers and stage directors can benefit from the interactive capability of our scheme.

Our scheme can handle rhythmic motions beats of which match those of background music. Moreover, the beats of such a motion should have the same fixed period. Some kinds of motions may have different meters than that of background music. For example, a *Swing* dance includes the basic movements of the $6/4$ meter while the background music has the $4/4$ meter. Our scheme cannot handle such a motion without *a priori* knowledge on the motion type. Automatic recognition of the meter of a basic movement without such knowledge will be a good future research topic on rhythmic motion analysis.

We are planning to extend our scheme to reflect the continuity between motion phrases, each of which consists of multiple basic movements. For automated music-driven choreography, we will also extend the idea of learning the transitions among the basic movements to learning the background music appropriate to those movements.

Acknowledgements

We appreciate the anonymous reviewers and referees for their constructive comments and suggestions that improved the paper. We would like thank Min Gyu Choi and Minje Park for fruitful discussions. We are grateful to Digital Agents, Inc. and nXEN Technology Co. for providing character models and captured motion data. This research was supported by the National Research Laboratory Program of the Ministry of Science & Technology, and the Brain Korea 21 Program of the Ministry of Education & Human Resources Development.

References

ARIKAN, O., AND FORSYTH, D. A. 2002. Interactive Motion Generation from Examples. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002) 21*, 3 (July), 483–490.

BEZDEK, J. C. 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York.

BINDIGANAVALE, R., AND BADLER, N. I. 1998. Motion Abstraction and Mapping with Spatial Constraints. In *Proc. International Workshop, CAPTECH '98*, 70–82.

BLOM, L., AND CHAPLIN, L. T. 1982. *The Intimate Act of Choreography*. Pittsburgh: University of Pittsburgh Press.

BOTTOMER, P. 1998. *Let's Dance!* Black Dog & Leventhal Publishers.

BRAND, M., AND HERTZMANN, A. 2000. Style Machines. *Computer Graphics (Proc. SIGGRAPH 2000) 34* (July), 183–192.

BRUDERLIN, A., AND WILLIAMS, L. 1995. Motion Signal Processing. *Computer Graphics (Proc. SIGGRAPH '95) 29* (Aug.), 97–104.

CASSELL, J., VILHJÁLMSSON, H. H., AND BICKMORE, T. 2001. BEAT: the Behavior Expression Animation Toolkit. *Computer Graphics (Proc. SIGGRAPH 2001) 35* (Aug.), 477–486.

CHIU, S. 1994. Fuzzy Model Identification Based on Cluster Estimation. *Journal of Intelligent & Fuzzy Systems 2*, 3 (Sept.).

DUBES, R., AND JAIN, A. K. 1976. Clutering Techniques: The User's Dilemma. *Pattern Recognition 8*, 247–260.

ECK, D., GASSER, M., AND PORT, R. 2000. Dynamics and Embodiment in Beat Induction. In *Rhythm Perception and Production*, 157–170.

ESSENS, P. J., AND POVEL, D. 1985. Metrical and Nonmetrical Representation of Temporal Patterns. *Perception and Psychophysics 37*, 1–7.

FOGEL, E., AND GAVISH, M. 1988. Parameter Estimation of Quasiperiodic Sequences. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, 2348–2351.

FORGY, E. 1965. Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classifications. *Biometrics 21*, 768.

GALATA, A., JOHNSON, N., AND HOGG, D. 2001. Learning Variable-Length Markov Models of Behavior. *Computer Vision and Image Understanding 81*, 398–413.

GLEICHER, M. 1998. Retargetting Motion to New Characters. *Computer Graphics (Proc. SIGGRAPH '98) 32* (July), 33–42.

GOMES, J., AND VELHO, L. 1999. From Fourier Analysis to Wavelets. In *Course Notes #5 (SIGGRAPH '99)*.

GUO, S., AND ROBERGÉ, J. 1996. A High-level Control Mechanism for Human Locomotion Based on Parametric Frame Space Interpolation. In *Proc. Eurographics Workshop on Computer Animation and Simulation 96*, 97–107.

HAWKINS, A. 1988. *Creating through dance*, revised ed. Princeton.

HELBING, D., AND MOLNAR, P. 1995. Social force model for pedestrian dynamics. *Physical Review E 51*, 4282–4286.

HELBING, D., FARKAS, I. J., AND T.VICSEK. 2000. Simulating dynamic features of escape panic. *Nature 407*, 487–490.

HUBER, P. J. 1981. *Robust Statistics*. John Wiley & Sons, New York.

J. VAN CAMP. 1994. Copyright of Choreographic Works. *1994–1995 Entertainment, Publishing and the Arts Handbook*, 59–92.

JONES, M. R., AND BOLTZ, M. 1989. Dynamic Attending and Responses to Time. *Psychological Review 96*, 3, 459–491.

KIM, D., KIM, H. K., AND SHIN, S. Y. 2003. An Event-driven Approach to Crowd Simulation with Example Motions. Tech. Rep. CS-TR-2002-186, Dept. Comput. Sci., KAIST, Taejon, Korea (Republic of).

KIM, M. J. 1996. *General Schemes for Unit Quaternion Curve Construction*. PhD thesis, *DCS88343*, Department of Computer Science, KAIST.

KIM, H. K. 2002. *Efficient Collision Detection Among Moving Spheres with Unknown Trajectories*. PhD thesis, *DCS02009*, Department of Computer Science, KAIST.

KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion Graphs. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002) 21*, 3 (July), 473–482.

LARGE, E. W., AND KOLEN, J. F. 1994. Resonance and the Perception of Musical Meter. *Connection Science 6*, 2/3, 177–208.

LAYBOURNE, K. 1998. *The Animation Book*. Three Rivers Press.

LEE, J., AND SHIN, S. Y. 1999. A Hierarchical Approach to Interactive Motion Editing for Human-like Figures. *Computer Graphics (Proc. SIGGRAPH '99) 33* (Aug.), 395–408.

LEE, J., CHAI, J., REITSMA, P. S. A., HODGINS, J. K., AND POLLARD, N. S. 2002. Interactive Control of Avatars Animated with Human Motion Data. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002) 21*, 3 (July), 491–500.

LI, X., AND PARIZEAU, M. 2000. Training Hidden Markov Models with Multiple Observations – A Combinatorial Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 4 (Apr.), 371–377.

LI, Y., WANG, T., AND SHUM, H. 2002. Motion Texture: A Two-Level Statistical Model for Character Motion Synthesis. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002) 21*, 3 (July), 465–472.

MINTON, S. C. 1997. *Choreography: A Basic Approach Using Improvisation*, 2nd ed. Human Kinetics.

PARK, S. I., SHIN, H. J., AND SHIN, S. Y. 2002. On-line Locomotion Generation Based on Motion Blending. In *Proc. ACM SIGGRAPH Symposium on Computer Animation*, 105–111.

PAUSCH, R., SNODDY, J., TAYLOR, R., WATSON, S., AND HASELTINE, E. 1996. Disney's Aladdin: First Steps Toward Storytelling in Virtual Reality. *Computer Graphics (Proc. SIGGRAPH '96) 30* (Aug.), 193–203.

POPOVIĆ, Z., AND WITKIN, A. 1999. Physically-based Motion Transformation. *Computer Graphics (Proc. SIGGRAPH '99) 33*, 11–20.

PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., AND FLANNERY, B. P. 1999. *Numerical Recipes in C – The Art of Scientific Computing*, second ed. Cambridge University Press.

PROAKIS, J. G., AND MANOLAKIS, D. G. 1996. *Digital Signal Processing: Principles, algorithms, and applications*, third ed. Prentice Hall International Editions.

PULLEN, K., AND BREGLER, C. 2002. Motion Capture Assisted Animation: Texturing and Synthesis. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002) 21*, 3 (July), 501–508.

RABINER, L. R. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. the IEEE 77*, 2 (Feb.), 257–286.

ROSE, C., GUENTER, B., BODENHEIMER, B., AND COHEN, M. F. 1996. Efficient Generation of Motion Transitions using Spacetime Constraints. *Computer Graphics (Proc. SIGGRAPH '96) 30* (Aug.), 147–154.

ROSE, C., BODENHEIMER, B., AND COHEN, M. 1998. Verbs and Adverbs: Multidimensional Motion Interpolation. *IEEE Computer Graphics and Applications 18*, 5, 32–40.

SADLER, B. M., AND CASEY, S. D. 1998. On Periodic Pulse Interval Analysis with Outliers and Missing Observations. *IEEE Transactions on Signal Processing 46*, 11 (November), 2990–3002.

SCARBOROUGH, D., MILLER, P., AND JONES, J. A. 1992. On the perception of Meter. In M. Balaban, K. Ebcioglu, and O. Laske (Eds.) *Understanding Music with AI: Perspectives in Music Cognition*, Cambridge:MIT Press, 427–447.

SCHÖDL, A., SZELISKI, R., SALESIN, D. H., AND ESSA, I. 2000. Video Textures. *Computer Graphics (Proc. SIGGRAPH 2000) 34* (July), 489–498.

SHIN, H. J., LEE, J., GLEICHER, M., AND SHIN, S. Y. 2001. Computer Puppetry: An Importance-Based Approach. *ACM Transactions On Graphics 20*.

SHOEMAKE, K. 1985. Animating Rotation with Quaternion Curves. *Computer Graphics (Proc. SIGGRAPH '85) 19* (Aug.), 245–54.

SILVESTER, V. 1993. *Modern Ballroom Dancing*, new rev. ed. Trafalgar Square.

SLOAN, P., ROSE, C. F., AND COHEN, M. F. 2001. Shape by Example. In *Proc. of 2001 ACM Symposium on Interactive 3D Graphics*, 135–144.

TANCO, L. M., AND HILTON, A. 2000. Realistic Synthesis of Novel Human Movements from a Database of Motion Capture Examples. In *Proc. the IEEE Workshop on Human Motion*, 137–142.

UNUMA, M., ANJYO, K., AND TAKEUCHI, R. 1995. Fourier Principles for Emotion-based Human Figure Animation. *Computer Graphics (Proc. SIGGRAPH '95) 29* (Aug.), 91–96.

WILEY, D., AND HAHN, J. K. 1997. Interpolation Synthesis of Articulated Figure Motion. *IEEE Computer Graphics and Applications 17*, 6, 39–45.

WITKIN, A., AND POPOVIĆ, Z. 1995. Motion Warping. *Computer Graphics (Proc. SIGGRAPH '95) 29* (Aug.), 105–108.